B.Tech Project Report

Wire Detection using Computer Vision

Department of Mechanical Engineering

Submitted by

Abhishek Singh (B20ME005) & Udit Agarwal (B20ME076)

Under the supervision of

Dr. Riby A. Boby & Dr. Binod Kumar

August 2023 - November 2023

Indian Institute of Technology Jodhpur

NH 62, Karwar, Rajasthan, India

342030

Acknowledgments

We would like to express our sincere gratitude to the following individuals for their invaluable contributions to this project:

- **Dr. Riby Abraham Boby and Dr. Binod Kumar:** Our mentors, for their guidance, support, and encouragement throughout the project.
- **Our project partner:** For his collaboration, dedication, and teamwork.
- **Gaurav Sangwan and Devansh Menaria:** Our friends, for their assistance in data collection and experimentation.

We are also grateful to our families and friends for their unwavering support and encouragement. This project would not have been possible without the contributions of these individuals. We are truly grateful for their help and support.

Index

Abstract

The project introduces a approach to wire detection for aerial vehicles, leveraging advanced deep learning algorithms to address the challenges posed in detection of slender, visually similar wires. The transformative methodology, rooted in safety enhancement for aerial vehicles, involves a meticulous literature review, high-quality dataset curation, and model training utilizing the YOLOv8 architecture along with catenary curve fitting enabling real-time wire reconstruction and proactive obstacle avoidance. The YOLOv8 model, chosen for its efficiency in wire detection, underwent rigorous training and was successfully implemented on a Raspberry Pi, showcasing the feasibility of edge device deployment. The system's efficacy was validated through rigorous testing, marking a significant advancement at the intersection of deep learning, computer vision, and robotics.

1 Introduction

Wire detection is hindered by challenges like image noise, material/color variations, complex backgrounds, occlusion, scale/orientation changes, limited training data, and the demand for accurate yet computationally efficient models. Clutter, similar textures, and lighting variations add complexity. Factors like blurred images, post-processing issues, and environmental conditions exacerbate the difficulty.

1.1 Objective

Wire detection is a multifaceted solution that contributes to aviation safety, regulatory compliance, infrastructure protection, UAV operations, cost savings, public safety, and the uninterrupted provision of essential services. Wire detection helps pilots and operators take preventive measures, avoiding wire strikes that could lead to accidents and fatalities. Accidental contact with power lines or communication cables leads to disruptions in services, power outages, and damage to the infrastructure itself. Hence, its implementation is crucial for minimizing risks and optimizing the safety and efficiency of various activities involving low-altitude flight and infrastructure inspection.

Figure 1: Helicopter striking a Power Lines(left) & Air tanker strikes power line(right)

This project addresses the critical issue of wire detection and avoidance by introducing novel innovations in three key areas: wire segmentation, reconstruction, and avoidance. The task is inherently challenging due to the slender nature of wires, their potential occurrence at various orientations and locations, and the difficulty in distinguishing them from visually similar lines and edges. To overcome these challenges, we leverage recent advancements in deep learning, treating wire detection as a semantic segmentation task.

Our methodology extends beyond detection to include the estimation of 3D coordinates through the application of camera calibration techniques. By obtaining these 3D coordinates, we further enhance our approach by predicting the shape of the wire through the derivation of catenary equations. This comprehensive pipeline enables the reconstruction of wire geometry, providing a more accurate representation of the obstacle. Importantly, the entire process can be implemented on edge devices, ensuring real-time applicability and efficiency.

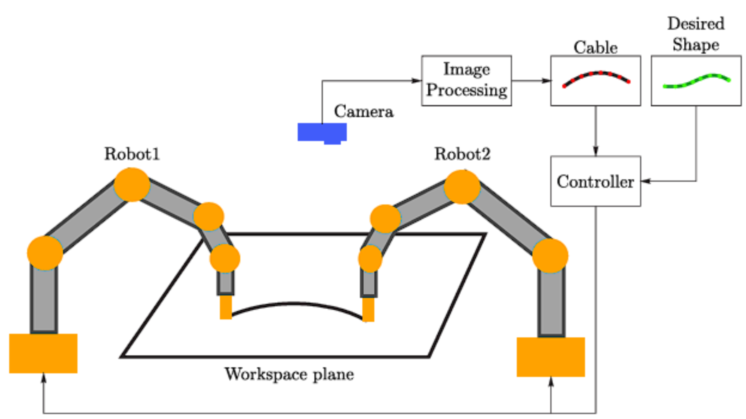
The primary objective of our project is to equip aerial vehicles with the capability to not only detect wires but also to reconstruct their spatial configurations. This spatial understanding enables the prediction of wire shapes, thereby facilitating proactive measures for obstacle avoidance. Beyond the immediate applications of UAV navigation, our methodology extends to tasks such as power cable inspections, broadening the scope of its practical utility.

This research is positioned at the intersection of computer vision, robotics and deep learning, presenting a holistic solution to a persistent challenge in the field of autonomous aerial systems. The successful implementation of our approach holds the potential to significantly reduce accidents, enhance the safety of UAV operations, and broaden the spectrum of tasks that UAVs can effectively and safely undertake.

1.2 Relevance to Mechanical Domain

The wire shape estimation that we are trying to achieve has multiple uses in the core mechanical domain like in the field of robotics (cable robots), UAVs and etc.

Deformable linear objects(DLOs), such as cables, ropes, and sutures, are involved in innumerable everyday life scenarios, such as cable management in industry or at home, and thread packing.



The wire equation can be used to accurately measure the length of cables in cable-driven robots and hence improve their precision.

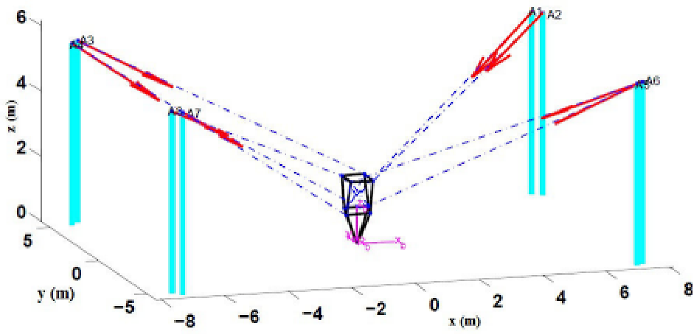


Figure 2: Sketch of the LIRMM/Tecnalia CoGiRo robot

1.3 Existing Methods

Researchers have explored various technologies for wire detection, including computer vision, LiDAR (Light Detection and Ranging), radar, and infrared imaging.

- Computer vision techniques involve analyzing images or videos to identify and track wires. One approach is to use edge detection algorithms to find the edges of wires, which are typically thin and linear. Another approach is to use machine learning algorithms, such as convolutional neural networks, to learn to classify pixels as either wire or non-wire.
- LiDAR is a laser-based technology that measures distances to objects. It is used to detect wires by creating a 3D point cloud of the environment and then identifying wire-like structures in the point cloud.
- Radar is a radio-based technology that can detect objects based on their reflections of radio waves. It is used to detect wires by identifying radar echoes that are consistent with the size and shape of wires.
- Infrared imaging is a technology that detects wires based on their heat signature. It is particularly useful for detecting wires that are obscured by vegetation or other objects.
- Sensor fusion is an approach that combines data from multiple sensors to improve wire detection accuracy. For example, a system that combines data from a camera and a LiDAR sensor use the camera to identify wire-like structures in images and then use the LiDAR sensor to confirm the presence of wires.

1.4 Literature Review

- ”Wire Detection using Synthetic Data and Dilated Convolutional Networks for Unmanned Aerial Vehicles” (2018). The approach in the paper is to use synthetic data and dilated convolutional networks for wire detection in unmanned aerial vehicles. The authors generate synthetic images of wires and use them to pretrain the network, which is then fine-tuned on real data. They also experiment with adding edge and line detection information as additional channels to the network input. The performance of their approach is compared to several baselines using standard evaluation metrics. The authors found that their method was able to achieve high accuracy in real-time on a portable GPU.

- Multi-view Reconstruction of Wires using a Catenary Model: The approach in the paper involves using a model-based approach to reconstruct wires with more fidelity by exploiting the natural physical structure of the problem, even in the presence of errors (due to occlusion or missed detections) in the wire segmentation process. The authors use a catenary model to estimate the wire model parameters via non-linear least squares optimization, and minimize the reprojection error using a computationally inexpensive distance transform as their loss function. The algorithm is designed to be efficient and robust, and can handle spurious (false positive) segments and predict the presence of wires despite missed (false negative) wire segments.
- Automatic High Resolution Wire Segmentation and Removal: The approach of the paper consists of a two-stage method for automatic wire segmentation and removal. The first stage involves segmenting wires in high-resolution images using a combination of global and local contexts. The second stage employs a tile-based inpainting strategy to remove the wires. The system also introduces a wire segmentation benchmark dataset to evaluate the performance of the approach.The paper shares a dataset of its test images which are of natural images from ground perspective and wire segmentation.

2 Work done

2.1 Methodology

Our comprehensive methodology aimed at advancing aerial vehicles safety through sophisticated wire detection and avoidance. Commencing with an in-depth literature review, we meticulously curated a high-quality dataset using Roboflow for effective model training. The model selection process identified YOLO as the most efficient architecture for wire detection, followed by an extensive training regimen. Camera calibration was crucial to derive intrinsic and extrinsic parameters, facilitating the conversion of 2D image coordinates into precise 3D representations. Utilizing these calibrated parameters, we estimated the 3D coordinates of identified wire points and applied a catenary equation for accurate shape prediction. Notably, the selected model was successfully implemented on a Raspberry Pi, showcasing the feasibility of edge device deployment. Rigorous testing validated the system's efficacy in wire detection and avoidance scenarios, marking a significant advancement at the intersection of deep learning, computer vision, and robotics. Result analysis and discussions yielded valuable insights for future enhancements and broader applications in the realm of UAV safety and navigation.

2.2 Implementation

2.2.1 Choosing the right model:

You only look once(YOLO) v8 is a powerful and versatile computer vision model that offers a number of advantages over other models. Yolov8 is lightweight as compared to others. For instance:

- Faster R-CNN : 30 million parameters
- Mask R-CNN : 40 million parameters
- Yolov8 : 13 million parameters

To test this model, we used a open-source dataset of 43 images available on the internet. It showed an impressive true-positive rate of 0.79. From here, we finalized Yolo as our method for wire detection. YOLOv8 is efficient and robust, and it achieved Mean Average Precision(mAP) score of 52.8% on the COCO dataset at a high speed of 60 FPS. Even in challenging scenarios, it maintains a strong performance:

- mAP score of 45.6% on the COCO dataset under low light conditions.
- mAP score of 47.8% on the COCO dataset in cluttered scenes.

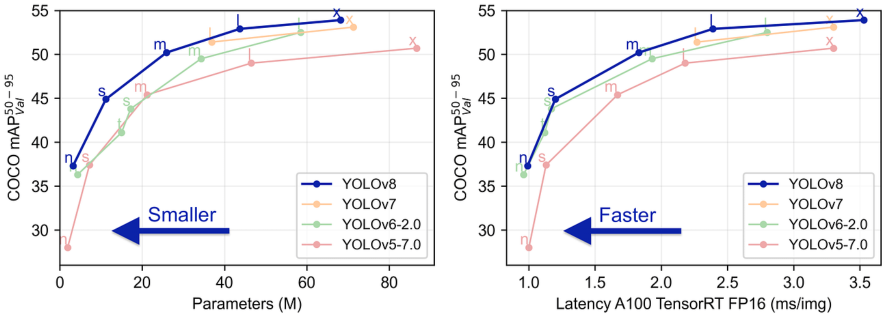


Figure 3: Yolov8

It is highly accurate, fast, scalable, easy to use, and open source. This makes it a good fit for our application.

2.2.2 Dataset Acquisition and Preprocessing

We require a very specific dataset for fine tuning our model such as images taken from the different altitude/angles of wires. Datasets consisting of images of cable robots, transmission lines and other types of cables hanging from two points was required. Very few publically available datasets were available; one such dataset that we were able to find was Transmission Towers and Power Lines (TTPLA) dataset. We used this dataset for detection and segmentation of transmission towers and power lines.
TTPLA is a public dataset which is a collection of aerial images on Transmission Towers (TTs) and Power Lines (PLs). It consists of 2,370 images with the resolution of (3840 × 2160) pixels.

Figure 4: Original Dataset

The next task we had was to label and annotate the images so that this can be fed to our segmentation model for training. For this purpose we used Roboflow for annotating the images into four classes: cable, tower_lattice, tower_tucohy, tower_wooden. After that we divided the dataset into Train, Validate and Test in the ratio 7:2:1. Since the dataset images had a high resolution, we had to resize them to (1280 x 1280) since that is enough. Anything above 1280 x 1280 would only cost us computation with negligible improvement in accuracy.

Figure 5: Annotation : image on the left annotated as image on right

Here the red highlighted area respresents wire and the blue one means wooden tower.

2.2.3 Model Training and Evaluation

We trained our dataset on two versions of Yolov8 namely nano and medium.

Model	size (pixels)	mAP^{box} 50-95	mAP^{mask} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)
YOLOv8n-seg	640	36.7	30.5	96.1	1.21	3.4
YOLOv8s-seg	640	44.6	36.8	155.7	1.47	11.8
YOLOv8m-seg	640	49.9	40.8	317.0	2.18	27.3
YOLOv8l-seg	640	52.3	42.6	572.4	2.79	46.0
YOLOv8x-seg	640	53.4	43.4	712.1	4.02	71.8

Figure 6: Different modes of YOLOv8 and their features

These selected YOLO model underwent rigorous training on the curated dataset. Training involved iterative adjustments to optimize the model's performance in wire detection. Evaluation metrics such as precision, segmentation loss, recall, and F1 score were employed to quantitatively assess the model's effectiveness in identifying wires accurately. We trained our model till we got near about constant segmentation loss.

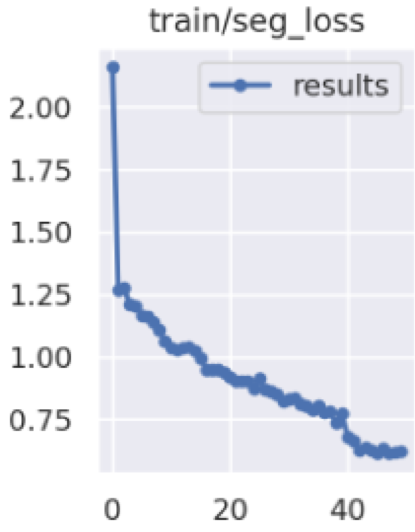


Figure 7: Training: Segmentation loss vs Number of Epochs

After about 43 epochs, the segmentaion loss was constant.

2.2.4 Camera Calibration

Post model selection and training, we proceeded to calibrate the camera to obtain the intrinsic and extrinsic parameters. This calibration step was crucial for converting 2D image coordinates into accurate 3D coordinates. The resulting camera matrix and distortion matrix played a pivotal role in enhancing the spatial understanding of detected wires.

2.2.5 3D Coordinate Estimation

Leveraging the calibrated camera parameters, we implemented a process to estimate the 3D coordinates of detected wire points. This step facilitated a more accurate representation of the wires in the real-world environment, laying the groundwork for subsequent analysis and reconstruction.

2.2.6 Catenary Equation Fitting

The obtained 3D coordinates were used to fit the catenary equation, a mathematical model describing the shape of wires under the influence of gravity. This step enabled us to predict and reconstruct the precise shape of the wire, contributing to a more nuanced understanding of the obstacle in the UAV's path.

2.2.7 Edge Device Implementation

To validate the real-world applicability of our solution, we implemented our image segmentation model on a Raspberry Pi.

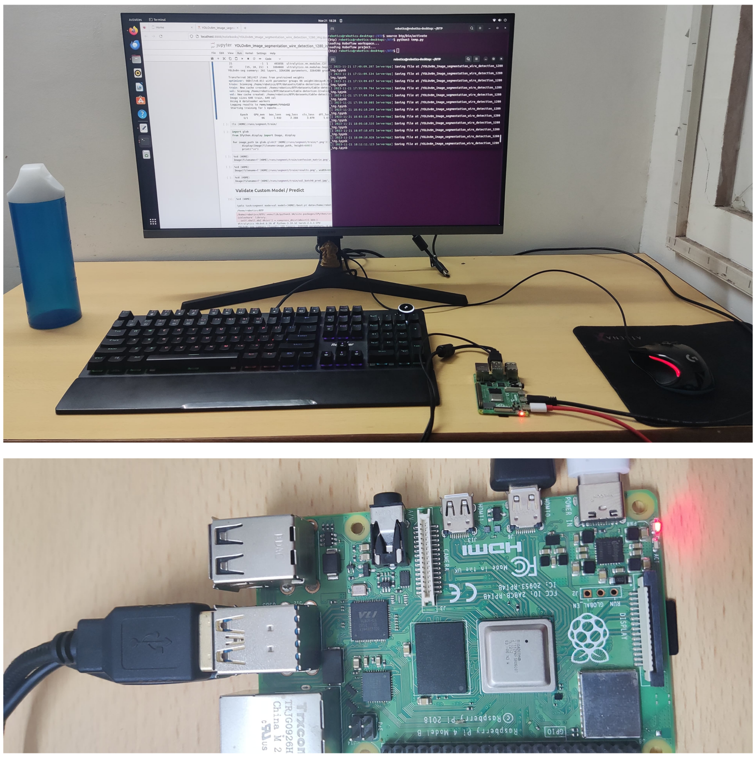


Figure 8: Implementation on Raspberry Pie

This step showcased the feasibility of deploying our model on edge devices, emphasizing its potential for integration into UAV systems with resource constraints.

2.3 Theoretical Work

2.3.1 Catenary Equation

The most basic form of the 2D catenary curve can be expressed with the following equation (Lockwood 1961):

$$y = c \cdot \cosh\left(\frac{x}{c}\right)$$

where c is a scaling factor that is governed by the ratio between the tension at the cable's vertices and the weight of the cable per unit length. The above equation can be extended further to include two translation factors **a** and **b**

$$y = a + c \cdot \cosh\left(\frac{x - b}{c}\right)$$

where a and b are the translation parameters from the origin along the y and x axis respectively. A wire with uniform density and thickness, when suspended by its two ends under uniform gravity, follows the shape of the planar, single-parameter, catenary curve.

There are different mathematical models for a3D catenary curve for conditioning the points lying on the curve during a least square adjustment of calibration

One famous and reliable model is as follows, a, b and c are functions of and b is a function of . Thus, and can be eliminated. In this model, the catenary orientation is implicitly modeled with the distance, u, from the centroid of the xy-coordinates of the set of points on the curve. It is known from regression that the centroid position lies on the best-fit straight line. The centroid,(xm, ym) is not estimated in the least-squares process but is updated in each iteration as the observations are updated in the the Gauss-Helmert adjustment mode:

$$z = a + c \cdot \left(\cosh\left(\frac{u-b}{c} \right) - 1 \right)$$

$$u = \pm\sqrt{(x - x_m)^2 + (y - y_m)^2}$$

2.3.2 Camera Calibration

Camera calibration is a process used to determine the intrinsic and extrinsic parameters of a camera. These parameters are essential for transforming a 2D image captured by the camera into real-world 3D coordinates. The intrinsic parameters include the focal length, principal point (the optical center of the image), and lens distortion coefficients, while the extrinsic parameters define the camera's position and orientation in 3D space.

In short, we need to find five parameters, known as distortion coefficients given by:

$$Distortion\ coefficients = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3)$$

Intrinsic parameters are specific to a camera. They include information like focal length (fx,fy) and optical centers (cx,cy). The focal length and optical centers can be used to create a camera matrix, which can be used to remove distortion due to the lenses of a specific camera. The camera matrix is unique to a specific camera, so once calculated, it can be reused on other images taken by the same camera. It is expressed as a 3x3 matrix:

$$camera\ matrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

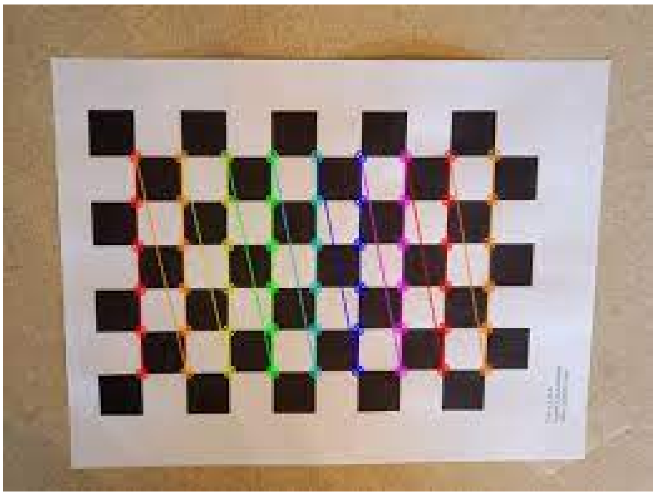


Figure 9: Webcam calibration using open-cv

2.3.3 Combined Pipeline

The combined pipeline can be uploaded on hardware (Raspberry Pi 4) and mounted on drones or industrial robots, with a camera and lidar sensor (for depth estimation). The segmentation mask values can be used with camera calibration values to predict the 3D coordinates. The estimated coordinates and depth values will be put into the catenary model to predict the values of catenary curve parameters (a,b,c). The obtained curve equation in live camera feed of for individual images can be used for multiple purposes.

3 Results and Discussion

The two models we trained performed quite well for our task.

Figure 10: Yolov8-Nano

Out of 100 instances where there was a wire, Our Yolov8-Nano model was able to detect 64 instances correctly.

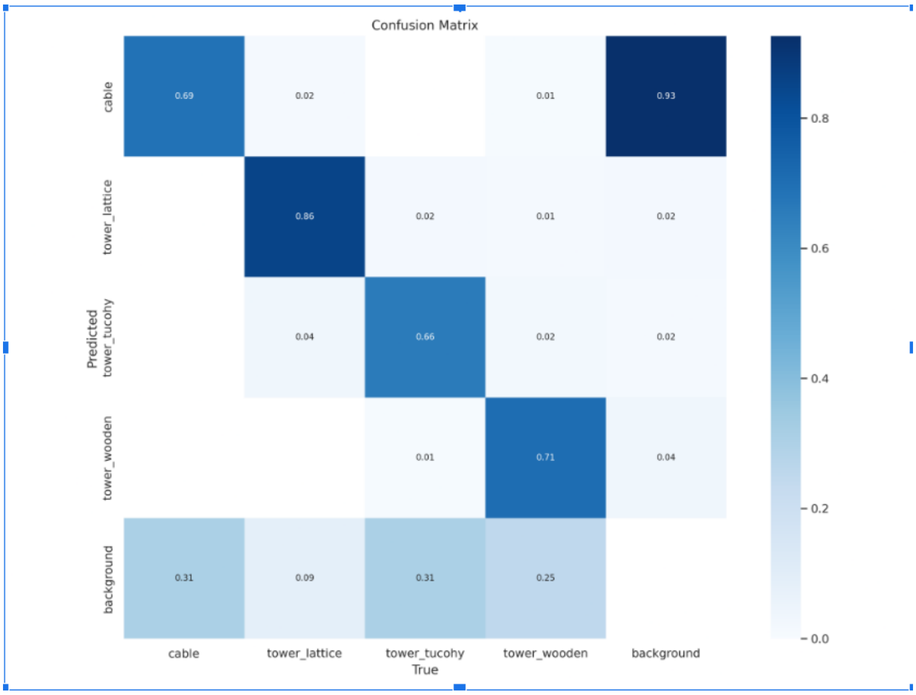


Figure 11: Yolov8-Medium

The results of our Yolov8-Medium model in wire detection demonstrate a promising performance, correctly identifying 69 out of 100 instances where wires were present. This indicates a considerable level of accuracy in detecting wires, showcasing the model's efficacy in this task. However, it's noteworthy that the Yolov8-Medium model, while more accurate than the Nano variant, comes with a higher computational cost, requiring 110 GFlops compared to Nano's 12.01 GFlops during training.

The trade-off between accuracy and computational efficiency is a common consideration in machine learning, and in this case, it's essential to weigh the benefits of improved detection against the increased computational demand. The discussion should delve into the practical implications of this trade-off, considering factors such as real-time application requirements, hardware constraints, and the overall objectives of the wire detection system.

Since our dataset was not balanced, this emphasizes the need to evaluate the model's performance beyond just accuracy. The discussion should address how the model handles this class imbalance, with a focus on the boxed average precision and recall metrics. A boxed average precision of 0.8 and a boxed average recall of 0.7 suggest a relatively balanced performance across precision and recall, indicating that the model is effective in both correctly identifying positive instances and minimizing false positives.

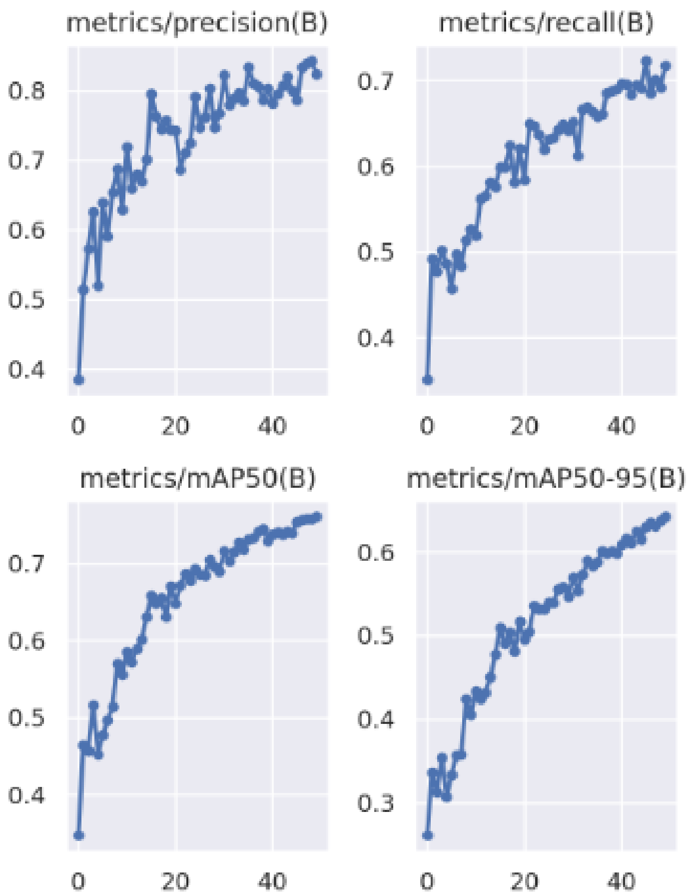


Figure 12: Boxed average metrics

On Nvidia T400 GPU, inference time per image of YOLOv8-medium was:

Process	**Time Taken**
Pre-process	1.4ms
Inference	27.7ms
Loss	0.0ms
Post-process	2.4ms
Total	**31.5ms**

On Nvidia T400 GPU, inference time per image of YOLOv8-nano was:

Process	**Time Taken**
Pre-process	1.3ms
Inference	11.4ms
Loss	0.0ms
Post-process	2.6ms
Total	**15.3ms**

The comparison of inference times between YOLOv8-medium and YOLOv8-nano on the Nvidia T400 GPU provides valuable insights into the trade-off between speed and performance. While YOLOv8-nano demonstrated faster inference times per image, completing the process in 15.3ms compared to YOLOv8-medium's 31.5ms, it's essential to consider the implications for overall model performance. The faster inference of YOLOv8-nano comes at the cost of reduced accuracy as evidenced by its confusion matrix during training. Furthermore, the breakdown of processing times into pre-process, inference, loss calculation, and post-process stages reveals the specific areas where each model excels. YOLOv8-nano exhibits efficiency in both the inference and overall processing stages, making it a suitable choice for applications where real-time responsiveness is critical. On the other hand, YOLOv8-medium, while slower, may offer enhanced detection accuracy and precision, making it more suitable for scenarios where exhaustive analysis is prioritized over speed.

The performance evaluation on the test dataset reinforces the discussion by providing real-world examples of how each model handles diverse images. This comprehensive analysis aids in selecting the most appropriate model based on the specific requirements of the application, considering the delicate balance between speed and accuracy. To proceed further we choose Yolov8-medium model.

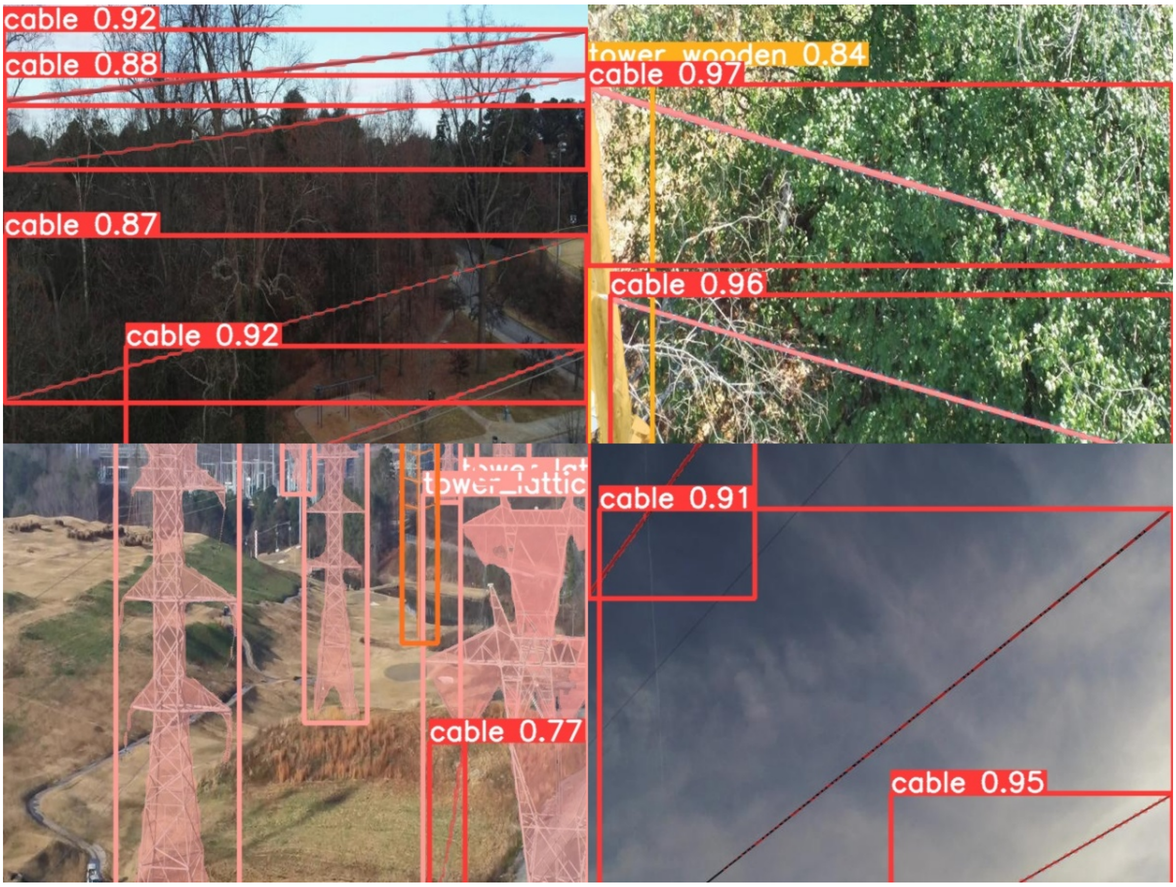


Figure 13: Predictions on Test Data

In the Output image, there are bounding box for each instance of class(wire and tower) with an probability score on the top left of box. Also the segmented instance is highlighted to differentiate it from the surrounding.

We then tried to run this model on raspberry pie 4 model B (1GB RAM).

Figure 14: Predictions on Raspberry pi

The wire are highlighted as blue On the Yolov8-medium model, we got an inference time of **2.2 seconds**. It is high due to several reason:

- Complex Model: Our model though lightweight still has 245 layers and 27224700(2̃7 million) parameters.
- Limited Hardware : The raspberry pi had only limited CPU without heatsink or the fans.
- Input Image size: The input image size was (3840 x 2160) . If we reduce the image size the inference time would decrease and the performance will also decrease.

Furthermore we even applied this model on images of cable-driven robots from the laboratory:

Figure 15: Predictions on indoor environment(lab)

4 Conclusion

Our project represents a significant stride in advancing the safety and capabilities of Aerial Vehicles (UAVs) through the development of a comprehensive wire detection and avoidance system. Our approach is better than previous works in multiple ways, we do not rely on multi-view reconstruction techniques which require a very specific dataset and are difficult to create and obtain. The obtained pipeline can be used to implement on-edge devices hence it can be used with UAVs, robotics arms, and other autonomous devices. The applied approach simply uses more publically available datasets hence there are fairly more future prospects of improvement in the accuracy and latency of the pipeline. This project not only contributes to the intersection of deep learning, computer vision, and robotics but also presents a holistic solution to the persistent challenge of navigating UAVs safely in environments containing thin obstacles such as wires or working robots in industries containing thin wires. The results obtained, coupled with insightful analyses and discussions, provide a foundation for future enhancements and applications in UAV and robots safety and navigation.

Appendix

- A1 Annotated Dataset on Roboflow
- A2 Github link for the repository
- A3 Colab Notebook Containing our work

5 References

1. Ratnesh Madaan, Michael Kaess, Sebastian Scherer, *Multi-view Reconstruction of Wires using a Catenary Model*, *2019 International Conference on Robotics and Automation (ICRA)*.
2. Mang Tik Chiu, Xuaner Zhang, Zijun Wei, Yuqian Zhou, Eli Shechtman, Connelly Barnes, Zhe Lin, Florian Kainz, Sohrab Amirghodsi, Humphrey Shi, *Automatic High Resolution Wire Segmentation and Removal*, *CVPR*.
3. Rabab Abdelfattah, Xiaofeng Wang, and Song Wang, *TTPLA : An Aerial-Image Dataset for Detection and Segmentation of Transmission Towers and Power Lines*, *ACCV*, 2020.
4. T.-O. Chan and D. D. Lichti. *3D CATENARY CURVE FITTING FOR GEOMETRIC CALIBRATION*, *The International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences*.
5. Ratnesh Madaan, Daniel Maturana, Sebastian Scherer *Wire detection using synthetic data and dilated convolutional networks for unmanned aerial vehicles*, *2017 IROS)*.